УДК 519.2

О НУМЕРАЦИИ ПЕРЕСТАНОВОК И СОЧЕТАНИЙ ДЛЯ ОРГАНИЗАЦИИ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ В ЗАДАЧАХ ПРОЕКТИРОВАНИЯ УПРАВЛЯЮЩИХ СИСТЕМ

Н.Е. Тимошевская

Томский государственный университет
E-mail: timnat@mail.isc.tsu.ru

Предложен метод параллельной генерации таких комбинаторных объектов, как перестановки и сочетания. В основе метода лежит возможность нумерации перечисляемых объектов так, что по номеру можно легко восстановить сопоставленный ему объект. Приведены формулы вычисления номера заданного объекта (перестановки, сочетания) и алгоритмы построения объекта по его номеру. Представлены результаты экспериментов, подтверждающие эффективность метода.

В проектировании управляющих систем возникает необходимость в генерации таких комбинаторных объектов, как перестановки и сочетания. Например, в задачах проектирования дискретных схем на базе программируемых больших интегральных схем с матричной структурой возникает задача генерирования всех допустимых в некотором смысле подмножеств заданного множества. Их генерацию можно вести параллельно на вычислительных узлах многопроцессорной системы. Для этого требуется задать нумерацию перечисляемых объектов так, что по номеру можно легко восстановить сопоставленный ему объект. Построенная таким образом нумерация позволяет разбить все множество объектов на классы, мощности которых отвечают производительности процессоров в системе, и генерировать объекты различных классов параллельно на соответствующих процессорах. Предлагаются формулы вычисления номера заданного объекта (перестановки, сочетания) и алгоритмы построения объекта по его номеру.

Генерация перестановок

Рассматриваются перестановки $(a_1,...,a_n)$ из элементов множества $N=\{1,2,...,n\}$, т.е. перестановки степени n. Известен алгоритм генерации (перечисления) всех перестановок в лексикографическом порядке [1], т.е. так, что перестановка $(a_1,...,a_n)$ предшествует перестановке $(b_1,b_2,...,b_n)$, если для некоторого i, $1\le i<n$, имеет место $a_1=b_1$, $a_2=b_2$,...,$a_i=b_i$ и $a_{i+1}<b_{i+1}$. Генерация выполняется путем преобразования текущей перестановки к следующей. Например, перестановки степени 3 перечисляются в следующем порядке: 123, 132, 213, 231, 312, 321.

Назовем *индексом перестановки* ее порядковый номер I в последовательности всех перестановок степени n, расположенных в лексикографическом порядке, $1\le I\le n!$. Рассматриваются задача вычисления индекса заданной перестановки и обратная ей – задача построения перестановки по ее индексу. Первую можно решить путем перечисления и последовательной нумерации перестановок до тех пор, пока не будет получена заданная перестановка, что крайне неэффективно. Предлагается вычислять индекс I перестановки $(a_1,...,a_n)$ по следующей формуле

$$I=(\sum_{i=1}^{n-2}(a'_i-i)(n-i)!)+(a'_{n-1}-(n-2)),$$

где $a'_i=a_i+k$ и k – есть количество элементов a_j таких, что $a_j>a_i$ и $j<i$.

Пример. Пусть $n=4$, $(a_1,...,a_4)=(2,3,1,4)$. Тогда: $(a'_1,a'_2,a'_3,a'_4)=(2,3,3,4)$, $I=(2-1)3!+(3-2)2!+(3-2)=9$.

Покажем каким образом перестановку можно построить по ее индексу. Пусть $0\le t<n$ и $A=(a_1, ..., a_t)$ есть префикс перестановки с индексом I. Обозначим J_t число перестановок, предшествующих всем перестановкам с префиксом A, и положим $M_t=I-J_t$. Назовем M_t *кратностью префикса A*. По определению $M_0=I$ и M_t есть число перестановок с префиксом A, индексы которых не превосходят I. Пусть $Y=\{y_1,...,y_{n-t+1}\}=N-\{a_1,...,a_{t-1}\}$ и $y_1<...<y_{n-t+1}$. Тогда по свойству перестановки $a_t\in Y$. Допустим, что $a_t=y_k$, $1\le k\le n-t+1$. Тогда $A=(a_1,...,a_{t-1},y_k)$ и $J_t=J_{t-1}+(k-1)(n-1)!$, поэтому $M_t=I-J_t=I-J_{t-1}-(k-1)(n-1)!=M_{t-1}-(k-1)(n-t)!$. Таким образом, зная индекс перестановки и любой ее префикс, можно вычислить кратность последнего.

Теорема 1. Пусть $1\le t<n-1$, $(a_1,...,a_{t-1})$ – префикс перестановки с индексом I и M_{t-1} – его кратность. Пусть также $\{y_1,...,y_{n-t+1}\}=N-\{a_1,...,a_{t-1}\}$ и $y_1<...<y_{n-t+1}$. Тогда t-й элемент a_t в перестановке с индексом I есть y_k для

$$k=\left\lceil\frac{M_{t-1}+(n-t)!-1}{(n-t)!}\right\rceil.$$

Доказательство. Пусть C_k для $1\le k\le n-t+1$ есть множество всех перестановок с префиксом $A=(a_1,...,a_{t-1},y_k)$. Число перестановок в нем $|C_k|=(n-t)!$. Ввиду $y_k<y_{k+1}$ все перестановки в C_k предшествуют всем перестановкам в C_{k+1}. Множества $C_1,C_2, ...,C_{n-t+1}$ образуют разбиение множества всех перестановок с префиксом $(a_1,...,a_{t-1})$, и перестановка с индексом I входит в одно из них. Пусть J обозначает число всех перестановок, предшествующих перестановкам с этим префиксом. Тогда $M_{t-1}=I-J$, и индексы всех перестановок в C_k суть $J+(k-1)(n-t)!+1$, $J+(k-1)\times\times(n-t)!+2,...,J+k(n-t)!+1$. При $(k-1)(n-t)!+1\le\le M_{t-1}\le k(n-t)!$ перестановка с индексом I входит в C_k и $a_t=y_k$. Прибавив ко всем частям последнего неравенства величину $(n-t)!-1$, получим $(k-1)(n-t)!++(n-t)!\le M_{t-1}+(n-t)!-1\le k(n-t)!+(n-t)!-1$, или $k(n-t)!\le\le M_{t-1}+(n-t)!-1<(k-1)(n-t)!$, откуда

$$k\le\frac{M_{t-1}+(n-t)!-1}{(n-t)!}<k+1.$$

Следовательно, $k = \left[\dfrac{M_{t-1} + (n-t)! - 1}{(n-t)!} \right]$.

Теорема 1 доказана.

При известном префиксе $(a_1,\ldots,a_{n-1})$ перестановки с индексом I ее последний член находится как единственный элемент множества $N-\{a_1,\ldots,a_{n-1}\}$.

Пример. Пусть n=4. Построим перестановку (a_1, a_2, a_3, a_4) по индексу I=9.

t=1: M_0=9, Y={1, 2, 3, 4}, k=2; следовательно, a_1=2.

t=2: $M_1=M_0-(k-1)(n-1)!=3$; Y={1, 3, 4}, k=2; следовательно, a_2=3.

t=3: $M_2=M_1-(k-1)(n-2)!=1$; Y={1, 4}, k=1; следовательно, a_3=1.

t=4: Y={4}; следовательно, a_4=4.

В результате получаем перестановку (2, 3, 1, 4).

Данный метод вычисления перестановки $(a_1,\ldots,a_n)$ по ее индексу I можно выразить следующим алгоритмом.

Алгоритм 1 (вычисление перестановки $(a_1,\ldots,a_n)$ по индексу I)

1. t=1; $(y_1\, y_2 \ldots y_n)$= (1 2...n).
2. Если t=n, то п. 5; иначе $k = \left[\dfrac{I + (n-t)! - 1}{(n-t)!} \right]$ и
3. $a_t=y_k$; $(y_1\ldots y_{n-t})=(y_1\ldots y_{k-1}y_{k+1}\ldots y_{n-t+1})$;
4. $I=I-(k-1)(n-t)!$; $t=t+1$ перейти к п. 2.
5. $a_n=y_1$.

Реализация алгоритма 1 предполагает использование "больших" чисел (целых чисел, двоичное представление которых требует большего числа разрядов, чем в стандартных типах) для задания индекса. Вместе с тем, опираясь на него, можно предложить следующий алгоритм параллельной генерации перестановок.

Пусть требуется построить все перестановки степени n на однородной многопроцессорной вычислительной системе, состоящей из p одинаковых процессоров с общей или распределенной памятью. Максимальную загруженность процессоров можно обеспечить, разделив последовательность всех перестановок на блоки по $s=[n!/p]$ элементов в каждом. Блок с номером m начинается с перестановки, соответствующей индексу $I=s(m-1)+1$, и генерация перестановок в нем возлагается на m-й процессор. Если число всех перестановок не кратно p, то процессор, генерирующий перестановки последнего блока, строит также (n! mod p) оставшихся перестановок, число которых меньше p. При выполнении этого алгоритма отсутствует взаимодействие между процессорами, и все процессоры одинаково загружены, что позволяет добиться ускорения (величина, показывающая, во сколько раз параллельный алгоритм работает быстрее последовательного) близкого к p.

Описанный параллельный алгоритм обладает высокой эффективностью благодаря пропорциональному распределению вычислений и отсутствию обменов данными между процессорами. Предложенный алгоритм был реализован с помощью технологии параллельного программирования MPI (Message Passing Interface), предназначенной для организации параллельных вычислений в системах с распределенной памятью, и исследован в эксперименте на однородной многопроцессорной системе кластерного типа. На рис. 1 представлены результаты вычислительного эксперимента, из которых следует, что средняя эффективность использования системы алгоритмом равна 0,94.

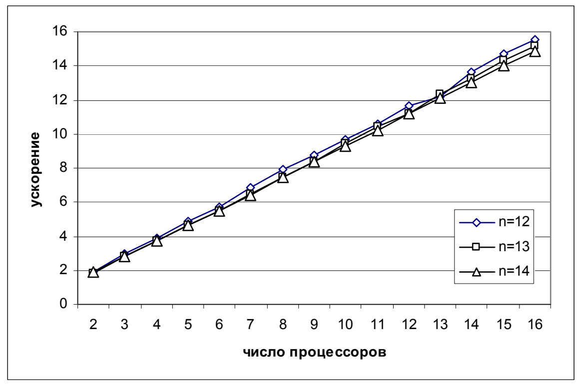


Рис. 1. *Ускорение для алгоритма параллельной генерации перестановок степени n, достигаемое при использовании 2, 3, ..., 16 процессоров*

Генерация сочетаний

Здесь под сочетанием понимается всякое упорядоченное по возрастанию k-элементное подмножество n-элементного множества N={1, 2, ..., n}, т.е. любой такой вектор $(b_1, b_2, \ldots, b_k)$ длины $k \le n$ с компонентами в N, называемый также сочетанием из n по k, в котором $b_i<b_{i+1}$ для i=1, ..., k–1. В [2] описан алгоритм генерации сочетаний, в котором сочетания порождаются в лексикографическом порядке, т.е. так, что сочетание $(b_1, b_2, \ldots, b_k)$ предшествует сочетанию $(a_1, a_2, \ldots, a_k)$, если для некоторого i, $1 \le i<k$, имеет место $b_1=a_1$, $b_2=a_2$, ..., $b_i=a_i$ и $b_{i+1}<a_{i+1}$. Например, все сочетания из 5 по 3 перечисляются в следующем порядке: (1, 2, 3), (1, 2, 4), (1, 2, 5), (1, 3, 4), (1, 3, 5), (1, 4, 5), (2, 3, 4), (2, 3, 5), (2, 4, 5), (3, 4, 5).

Индексом сочетания назовем порядковый номер I этого сочетания в последовательности всех сочетаний, расположенных в лексикографическом порядке, $1 \le I \le C_n^k$. Индекс I сочетания $(b_1, b_2, \ldots, b_k)$ из n по k может быть вычислен по следующей формуле:

$$I = 1 + \sum_{i=1}^{k} \sum_{j=b_{i-1}+1}^{b_i - 1} C_{n-j}^{k-i}, \quad \text{где } b_0 = 0 \text{ и } C_{n-j}^{0} = 1.$$

Пример. Вычислим индекс сочетания (2, 4, 5) из 5 по 3.

$$I = 1 + \sum_{j=0+1}^{2-1} C_4^2 + \sum_{j=2+1}^{4-1} C_2^1 + \sum_{j=4+1}^{5-1} C_0^0 = 1 + 6 + 2 + 0 = 9.$$

Рассмотрим теперь, как сочетание $(b_1,b_2,\ldots,b_k)$ строится по его номеру I. Пусть $1 \le t \le k$ и $0=b_0<b_1<b_2<\ldots<b_{t-1}$. Для любого j в $\{b_{t-1}+1, b_{t-1}+2, \ldots, n-k+t\}$, через s_j^t обозначим число всех сочетаний с

префиксами $(b_1,b_2,...,b_{t-1},r)$, где $r\in\{b_{t-1}+1,b_{t-1}+2,...,j-1\}$. В каждом из этих сочетаний недостающие $k-t$ элементов берутся из $n-r$ элементов $r+1,...,n$, поэтому $s_j^t=\sum_{r=b_{t-1}+1}^{j-1} C_{n-r}^{k-t}$. Положим также $s^t=\sum_{i=1}^{t-1} s_{b_i}^i$. Это есть число всех сочетаний с префиксами $(b_1,b_2,...,b_{i-1},a_i)$ для $i\le t-1$ и $a_i<b_i$, то есть всех сочетаний, предшествующих сочетаниям с префиксом $(b_1,b_2,...,b_{t-1})$.

Теорема 2. Пусть $1\le t\le k$ и $(b_1,b_2,...,b_{t-1})$ есть префикс сочетания с индексом I; тогда t-й элемент b_t этого сочетания равен наименьшему из таких j, что $b_{t-1}<j<n-k+t$ и $I\le s^t+s_{j+1}^t$, если такие j существуют, и $b_t=n-k+t$ в противном случае.

Доказательство. Для любого j, где $b_{t-1}<j<n-k+t$, введем $I_j=s^t+s_j^t$. По определению s^t и s_j^t есть число всех сочетаний, предшествующих сочетаниям с префиксом $(b_1,b_2,...,b_{t-1},j)$, т.е. I_j является индексом последнего сочетания с префиксом $(b_1,b_2,...,b_{t-1},j-1)$, а I_{j+1} – индексом последнего сочетания с префиксом $(b_1,b_2,...,b_{t-1},j)$. Рассмотрим возможные случаи, имея в виду, что $b_{t-1}<b_t\le n-k+t$.

1. Не существует такого j, что $b_{t-1}<j<n-k+t$. Тогда $b_t=n-k+t$.
2. Для всех j, где $b_{t-1}<j<n-k+t$, имеет место $I>I_{j+1}$. В этом случае $b_t>j$ для всякого j в границах $b_{t-1}<j<n-k+t$ и потому $b_t=n-k+t$.
3. Существуют j с ограничениями $b_{t-1}<j<n-k+t$ и $I\le I_{j+1}$. Для любого из них $b_t\le j$. Возьмем наименьшее j' из таких j. Тогда $I_{j'}<I$ и $b_t\ge j'$. Следовательно, $b_t=j'$.

Теорема 2 доказана.

Пример. Построим сочетание из 5 по 3 с индексом $I=9$.

$t=1$: $s^t=0$; $j=1$, $s_{j+1}^t=s_2^1=6$, $s^t+s_{j+1}^t=6<I$; $j=2$, $s_3^1=9$, $s^t+s_{j+1}^t=9\ge I$; следовательно, $b_1=2$.

$t=2$: $s^t=s_2^1=6$; $j=3$, $s_{j+1}^t=s_4^2=2$, $s^t+s_{j+1}^t=8<I$; $j=4=n-k+t$; следовательно, $b_2=4$.

$t=3$: не существует j для $4=b_{t-1}<j<n-k+t=5$, поэтому $b_3=n-k+t=5$.

В результате получаем сочетание (2, 4, 5).

Данный метод вычисления сочетания из n по k по его индексу I можно выразить следующим алгоритмом.

Алгоритм 2

1. $b_0=0$, $s=0$, $t=1$.
2. Если $t\le k$, то п. 3, иначе п. 8.
3. $j=b_{t-1}+1$.
4. Если $j<n-k+t$ и $s+C_{n-j}^{k-t}<I$, то п. 5, иначе п. 6.
5. $s=s+C_{n-j}^{k-t}$; $j=j+1$; п. 4.
6. $b_t=j$;
7. $t=t+1$; п. 2.
8. $(b_1, ..., b_k)$ – искомое сочетание.

Параллельная генерация сочетаний осуществляется так же, как и перестановок, а именно: все множество сочетаний делится на блоки, мощности которых отвечают производительности процессоров в системе; в каждом блоке по алгоритму 2 с помощью индекса вычисляется первое сочетание, начиная с которого один из процессоров системы генерирует сочетания всего блока. Как и в случае перестановок, благодаря пропорциональному распределению вычислений и отсутствию обменов данными между процессорами, параллельный алгоритм обладает высокой эффективностью.

Результаты эксперимента представлены на рис. 2, где изображен график усредненного значения ускорения, полученного в примерах со следующими параметрами: $n=30$, $k=15$; $n=40$, $k=10$; $n=40$, $k=15$; $n=40$, $k=30$; $n=50$, $k=10$; $n=50$, $k=40$; $n=100$, $k=6$; $n=100$, $k=94$. Средняя эффективность использования системы (отношение ускорения к числу процессов) равна 0,87.



Рис. 2. *Среднее ускорение для алгоритма параллельной генерации сочетаний, достигаемое при использовании 2, 3, ..., 16 процессоров*

Отметим, что в случае генерации сочетаний эффективность распараллеливания меньше, чем в случае генерации перестановок. Причина этого в том, что время, требуемое на построение блока сочетаний, зависит от того, какие именно сочетания в нем генерируются. В ходе эксперимента также было установлено, что процессоры, генерирующие сочетания первых блоков, заканчивают работу немного быстрее, чем процессоры, генерирующие сочетания последних блоков. Таким образом, при генерации сочетаний с большими индексами требуется, как правило, больше операций, и поэтому блоки одинакового размера могут строиться за различное время, однако для практических приложений эта разница не существенна.

Полученные в эксперименте значения коэффициентов эффективности 0,94 и 0,87, соответственно для алгоритмов генерации перестановок и сочетаний, подтверждают хорошую масштабируемость программ, то есть сохранение тенденции уменьшения времени работы при увеличении числа процессоров.

СПИСОК ЛИТЕРАТУРЫ

1. Дейкстра Э. Дисциплина программирования. – М.: Мир, 1978. – 280 с.
2. Липский В. Комбинаторика для программистов. – М.: Мир, 1988. – 200 с.